

Contents

| | |
|--|----------|
| 1 Routine/Function Prologues | 2 |
| 1.1 Fortran: Module Interface time_module.F90 (Source File: time_module.F90) | 2 |

1 Routine/Function Prologues

1.1 Fortran: Module Interface time_module.F90 (Source File: time_module.F90)

Module contains variables that represent LDAS time

REVISION HISTORY:

```
15 Oct 1999: Paul Houser; Initial Version
26 Apr 2000: Brian Cosgrove; Fix for Alpha (Used NINT instead
             of DINT b/c of rounding/truncating issues)
01 May 2000: Brian Cosgrove; Correction in TIME2DATE subroutine,
             if hour=24, hour set to 0, day, month and year advanced
```

INTERFACE:

```
module time_module
```

ARGUMENTS:

```
type listime
    integer :: sss          !starting second
    integer :: sdoy         !starting day of year
    integer :: smn          !starting minute
    integer :: shr          !starting hour
    integer :: sda          !starting day
    integer :: smo          !starting month
    integer :: syr          !starting year
    integer :: endcode      !0=realtim, 1=specific date
    integer :: ess          !ending second
    integer :: emn          !ending minute
    integer :: edoy         !ending day of year
    integer :: ehr          !ending hour
    integer :: eda          !ending day
    integer :: emo          !ending month
    integer :: eyr          !ending year
    integer :: ts            !timestep (seconds)
    integer :: tscount       !timestep count
    integer :: yyyyymmdd,hhmmss
    integer :: doy,yr,mo,da,hr,mn,ss !lis current model timing variables
    integer :: endtime        !lis stop (0=continue time looping)
    integer :: pda            !lis previous timestep day
    real*8 :: time           !lis current model time in years
    real*8 :: etime          !lis end time in years
    real :: gmt,egmt,sgmt
end type listime
```

Converts the date to year, month and day

INTERFACE:

```
subroutine date2days(time, yr, mo, da)
    implicit none
```

ARGUMENTS:

```
integer yr, mo, da
integer time
```

CONTENTS:

```
yr = time/10000
mo = (time-yr*10000)/100
da = (time-yr*10000-mo*100)
```

Converts time in seconds to hours, minutes and seconds

INTERFACE:

```
subroutine sec2time(sec, hr, mn, ss)
```

ARGUMENTS:

```
integer sec, hr, mn, ss
```

CONTENTS:

```
hr = sec/3600
mn = (sec-hr*3600)/60
ss = (sec-hr*3600-mn*60)
```

Updates the LDAS time variables according to the ESMF time

INTERFACE:

```
subroutine updatetime(lt)
    use lislog_module
    implicit none
```

ARGUMENTS:

```
type(listime)::lt
```

CONTENTS:

```
call date2time(lt%time, lt%doy, lt%gmt, &
    lt%yr, lt%mo, lt%da, lt%hr, lt%mn, lt%ss)
write(*,24)'GSFC-LIS time: ', lt%mo,'/', lt%da,'/', &
    lt%yr, lt%hr, ':', lt%mn, ':', lt%ss
write(79,24)'GSFC-LIS time: ', lt%mo,'/', lt%da,'/', &
    lt%yr, lt%hr, ':', lt%mn, ':', lt%ss
```

```

24 format(a16,i2,a1,i2,a1,i4,1x,i2,a1,i2,a1,i2)

    lt%yyyymmdd=(10000*lt%yr)+(100*lt%mo)+lt%da
    lt%hhmmss=(10000*lt%hr)+(100*lt%mn)+lt%ss
!-----
! check for endtime
!-----
    if(lt%endcode.eq.0)then !end at real-time date (tbd)
        write(*,*)'warning: do not know how to stop in real-time'
        write(79,*)'warning: do not know how to stop in real-time'
    endif
    if(lt%endcode.eq.1)then !end on date specified in lis.crd file
        call date2time(lt%etime,lt%edoy,lt%egmt, &
                      lt%eyr,lt%emo,lt%eda,lt%ehr,lt%emn,lt%ess)
        if(lt%time.ge.lt%etime)then
            lt%endtime=1
            write(*,*) 'GSFC-LDAS run completed'
            write(79,*)'GSFC-LDAS run completed'
        endif
    endif
    return

```

advance (or retract) time variables a specified amount (a nonmodular version of ticktime.f.)

REVISION HISTORY:

```

1 oct 1999: jared entin; initial code
15 oct 1999: paul houser; significant f90 revision

```

INTERFACE:

```

subroutine tick(time,doy,gmt,year,month,day,hour,minute,second)
    implicit none

```

ARGUMENTS:

```

real*8 time
integer days(13)
integer year,month,day,hour,minute,second,doy
real gmt

```

CONTENTS:

```

143 format(a1,' yr',i6,' mo',i5,' dy',i5,' hr',i5, &
          ' mn',i6,' ss',i8,' ts',i8)
ss=ss+ts

```

```
do while(ss.gt.59)
    ss=ss-60
    mn=mn+1
enddo
do while(ss.lt.0)
    ss=ss+60
    mn=mn-1
enddo
do while(mn.gt.59)
    mn=mn-60
    hr=hr+1
enddo

do while(mn.lt.0)
    mn=mn+60
    hr=hr-1
enddo
do while(hr.gt.23)
    hr=hr-24
    da=da+1
enddo

do while(hr.lt.0)
    hr=hr+24
    da=da-1
enddo

if((mod(yr,4).eq.0.and.mod(yr,100).ne.0) &           !correct for leap year
   .or.(mod(yr,400).eq.0))then                      !correct for y2k
    days(2)=29
else
    days(2)=28
endif

do while(da.gt.days(mo))
    da=da-days(mo)
    mo=mo+1
enddo

do while(da.lt.1)

    prvmo=mo-1
    if(mo.eq.1) prvmo=12

    da=da+days(prvmo)

    if(prvmo.eq.12) then
        mo=prvmo
```

```

        yr=yr-1
    else
        mo=prvmo
    endif
enddo
do while(mo.gt.12)
    mo=mo-12
    yr=yr+1
enddo

do while(mo.lt.1)
    mo=mo+12
    yr=yr-1
enddo
call date2time(time,doy,gmt,year,month,day,hr,mn,ss)
return

```

determines time in years, based on year, month, day hour etc.. or reverse (date2time).

REVISION HISTORY:

```

15 oct 1999: paul houser; initial code
21 feb 2002: brian cosgrove; corrected leap year code line. days(2)
              was not being reset to 28 after leaving a leap year,
              it was staying 29

```

INTERFACE:

```

subroutine date2time(time,doy,gmt,year,month,day,hr,mn,ss)

implicit none

```

ARGUMENTS:

```

integer year,month,day,hr,mn,ss,yrdays,doy,days(13),k
real*8 time
real gmt

```

CONTENTS:

```

if((mod(year,4).eq.0.and.mod(year,100).ne.0) &      !correct for leap year
   .or.(mod(year,400).eq.0))then                      !correct for y2k
    yrdays=366
else
    yrdays=365
endif

doy=0
do k=1,(month-1)

```

```
    doy=doy+days(k)
enddo
doy=doy+da

if(yrdays.eq.366.and.mo.gt.2)doy=doy+1

time=(dfloat(yr)+((((((dfloat(ss)/60.d0)+dfloat(mn))/60.d0)+ &
dfloat(hr))/24.d0)+dfloat(doy-1))/dfloat(yrdays))

gmt=( (float(ss)/60.0) +float(mn)) /60.0)+float(hr)
return
```